SYSTEMS AND METHODS FOR MINIMIZING HARMONIC INTERFERENCE

Inventor: Masoud Azmoodeh

BACKGROUND

This invention relates to systems and methods for minimizing harmonic in a communications circuit.

Frequency conversion is the process of changing one frequency to another. This may occur in instances when one signal is multiplied with a second signal to produce, a sum and/or difference of the signals. Frequency conversion circuits are commonly used in transmitters/receivers to convert an outgoing/incoming Intermediate Frequency/Radio Frequency (IF/RF) signal to the RF/IF signal. In case of receivers the IF signal is further processed by circuits following the frequency conversion circuit to develop a suitable signal for passing to a demodulator. The demodulator recovers the information encoded into the IF signal. The frequency conversion circuit produces the IF signal by mixing the RF signal with one or more local oscillator (LO) signals using a mixer.

Wireless receivers usually use several mixers to take the RF signal from the antenna and shift it to the lower IF frequency. This procedure usually will help on sampling the data and going to the digital domain. The frequency of the IF signal will be determined by the separation in frequency between the radio frequency signal and the local oscillator signals. The mixer receives separate input signals and combines the signals to produce an output signal. The mixer allows very high frequency to be down-converted to baseband or IF so that signals may be evaluated using analog or digital

signal processing techniques. In another example, the mixer up-converts a low frequency to a high frequency.

Potential detrimental interference can arise in the IF circuit. One form of detrimental interference includes harmonic interference, also referred to as "harmonic". Harmonics are generally spectral components of a signal that can interfere with proper operation of the wireless receiver. These digital transceiver ICs typically deploy many clocks in the same IC running at different frequencies. The problem becomes more pronounced for single chip digital RF transceivers since such single chip solution for wireless receivers requires combining the RF and Base Band processing. The effect of the harmonic of any of these clocks on the mixed signal part of the IC becomes significant. As an example of this problem, the third harmonic of a 1 MHz clock at the output of the last mixer appears as an IF signal at 3 MHz. This third harmonic of the 1 MHz clock will show up as a 3 MHz signal at the output of the IF mixer at 3 MHz and during the demodulation process this will look like a DC offset, which is undesirable.

SUMMARY

In one aspect, a method minimizes nth-order harmonic associated with a square wave clock signal having a predetermined frequency and a duty cycle by changing the duty cycle of the clock to eliminate or suppress the nth-order harmonic of the clock; and generating a low-interference clock having the changed duty cycle while keeping the predetermined frequency.

Implementations of the above aspect may include one or more of the following: Generating a low-interference clock further comprises generating an asymmetrical clock signal. Changing the duty cycle can change the position of the falling edge of the square wave clock relative to the position of the rising edge of the clock. Minimizing the nth-order harmonic changes the magnitude of the other harmonic. The resulting low-interference clock can be used in digital transceivers.

In another aspect, a system for minimizing nth-order harmonic associated with a square wave clock signal having a predetermined frequency and a duty cycle includes means for changing the duty cycle of the clock to eliminate or suppress the nth-order harmonic of the clock; and means for generating a low-interference clock having the changed duty cycle while keeping the predetermined frequency.

Implementations of the above aspect may include one or more of the following. The means for generating a low-interference clock further includes means for generating an asymmetrical clock signal. The means for changing the duty cycle further comprises means for changing the position of the falling edge of the square wave clock relative to the position of the rising edge of the clock. The Minimizing of the nth-order harmonic

changes the magnitude of other harmonic. The low-interference clock can be used in a digital radio transceiver.

In another aspect, a clock generator includes a clock oscillator; a down counter coupled to the clock oscillator; and a controller coupled to the down counter to generate a low harmonic clock with an asymmetrical duty cycle.

Implementations of the above aspect may include one or more of the following: The clock oscillator generates an output at a high frequency relative to the desirable low frequency clock rate. The counter is a modular down counter. The controller can change the position of the falling edge of the clock relative to the position of the rising edge of the clock. The controller can minimize the nth-order harmonic and changes the magnitude of other harmonic.

Advantages of the above system and method may include one or more of the following. The system reduces nth-order harmonic to almost zero without requiring the use of higher grade components or employing additional shielding around noisy components (through, for example, a metal shield to enclose the noisy device). The system achieves higher performance in a cost-effective manner.

The above system provides a method for suppressing adverse harmonic of the signal. Changing the duty cycle of the digital clock reduces or eliminates various specific harmonics. Additionally, one or more design parameters can be modified to reduce or eliminate unwanted harmonics. The system reduces a specific signal interference problem on a chip and minimizes the effect of the harmonics of the clocks on a chip on other parts of the chip.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows an exemplary square wave signal.

FIG. 2 shows an exemplary clock generator in accordance with the present invention.

FIG. 3 shows an exemplary process for generating clock signals with minimal interference.

FIG. 4 shows power spectrum charts for a 1KHz clock (symmetrical and asymmetrical).

FIG. 5 shows power spectrum charts for the 1KHz clock (symmetrical/asymmetrical).

FIG. 6 shows a time domain analysis of a 24 KHz clock along with the 1KHz clock with two different duty cycles.

DESCRIPTION

Fig. 1 shows an exemplary square wave $g(t)$ with period T and pulse width τ. The square wave can be described mathematically as follows:

$$g(t) = A, \quad -\tau/2 \le t \le \tau/2$$
$$= 0, \quad \text{For the remainder of the Period T.}$$

g(t) can be expanded using the Fourier transform:

$$g(t) = \sum_{-\infty}^{\infty} C_n \exp(j2\pi nt/T), \text{ where } C_n = 1/T \int_{-\tau/2}^{\tau/2} g(t)\exp(-j2\pi nt/T)dt.$$

The integral for C_n can be expressed as follows:

$$C_n = (A/n\pi).SIN(n\pi\tau/T), n = 0,\pm1,\pm2,\ldots.$$

As an example, for τ = T/2, (50-50 duty cycle), $|C_3|$ the magnitude of the third harmonic of the signal can be calculated as: $|C_3| = (A/3\pi).SIN(3\pi/2) = A/(3\pi)$ and for $\tau \neq T/2$, $|C_3| = (A/3\pi).SIN(3\pi\tau/T)$,

where $|SIN(.)| \le 1$.

Thus, for any value of $\tau \neq T/2$, the magnitude of the third harmonic is less than its value for a symmetric clock.

Using $\tau = T/3, C_3 = A/(3\pi).SIN(\pi) = 0$ and the third harmonic can be minimized.

As discussed next, the effect of third harmonic of 1MHz clock on the output of the IF mixer at 3 MHz is significantly reduced in one exemplary embodiment of the invention.

Figure 2 shows an exemplary clock generator. The system of Fig. 2 uses a relatively high frequency (in this example 24 kHz) clock/oscillator to generate an asymmetric low frequency clock (in this example a 1 KHz clock with 33/67 percent duty cycle). Figures 4-5 show the spectrum for both symmetric clock (50/50 percent duty cycle), and asymmetrical clock (33/67 percent duty cycle).

In the example of Figures4-5, the clock generator is arbitrarily set for the 33/67 percent duty cycle. However, any other duty cycle percentage can be used. As an example for generating the 33/67 percent duty cycle 1kHz clock as in Figure 2, counter1 counts from $(i-1)$ to 0, where i = 24 for the embodiment of Figure 2. Counter1 is a modular counter with modulo i and counts on the rising edge of the 24 KHz clock/oscillator.

During the system initialization (reset), control logic resets the counter1 to i. As long as $0 < counter1 \leq (j-1)$ where $j = 8$ control logic keeps the output signal at logic high, otherwise it will keep the output signal at logic low. This process is repeated and with appropriate parameters i & j, different clock rates with the desired duty cycle can be generated. Exemplary counter implementation in VERILOG is shown below.

```
******************************** DESCRIPTION ************************************
This module is a down counter that generates a clock with specific clock rate and
duty cycle using the input clock and parameters

Inputs:
        clk      : high speed input clock
        reset    : input reset signal
        val1     : 5 bit input, number of clk cycles that output clock will be high
        val2     : 5 bit input, number of clk cycles that output clock will be low
                 : val1 & val2 will generate the desired duty cycle and clock rate
Outputs:
        clk_out : output clock at the desired clock rate and duty cycle

****************************** Code **********************************************/

`timescale 1ns/100 ps

module downcntr (clk_out, clk, reset, val1, val2);
  //{
  output        clk_out;
  input         clk, reset, val1, val2;
```

```
//
wire   [4:0]  val1, val2;
wire          clk_out;
// local register
reg    [4:0]  count1;
//

// generate 64 KHz from 4 MHz clock
always @(posedge clk or posedge reset)
   begin //{
       if  (reset)             count1 <= val2-1;
       else if (count1 ==0)    count1 <= val2-1;
       else                    count1 <= count1 -1;
   end//}

assign clk_out = (count1 <=(val1-1));

//} end of the module
endmodule


********************************* DESCRIPTION ***************************************
This module is a testbench to test the down counter that generates a clock with specific clock
rate and duty cycle using the input clock and parameters.


Modules instantiated:
                     downcntr

******************************* Code ************************************************/
`timescale 1ns/100 ps

module tdowncntr;
  //{ begin the module
  parameter CLOCK = 41.6667;      // This is for 24 MHz clock
  // parameter CLOCK = 250;       // This is for  4 Mhz clock
  reg clk, rst;
  wire clk_o;
  reg  [4:0] val1, val2;

  downcntr clk_gen(
  .clk_out (clk_o),   // output, clock
  .clk     (clk),     // input,  high speed clock
  .reset   (rst),     // input
  .val1    (val1),    // input
  .val2    (val2)     // input
  );

  initial // Clock generator
    begin
      clk = 0;
      #80 forever #(CLOCK/2) clk = !clk;
    end

  // These two parameters are set for generating 1MHz clock from a 24 MHz clock with
  // duty cylce (1/3)/(2/3) (1/3 microsecond clock will be high, and 2/3 microsecond
  // clock will be low
  initial //set the parameters
    begin
      val1 = 5'd08;
      val2 = 5'd24;
    end
  //
  initial        // Test stimulus
    begin
      rst       = 0;
      #450  rst = 1;
      #400  rst = 0;
      #50000  $stop;
    end

  initial
      $monitor($stime,, rst,, clk,,,clk_o);

//} end of module
endmodule
```

Figure 3 shows an exemplary process that minimizes or eliminates nth-order harmonic, particularly nth-order harmonic associated with a square wave clock signal having a predetermined frequency and a duty cycle. The process includes changing the duty cycle of the clock to eliminate or suppress the nth-order harmonic of that clock (step 12). Next, a low-interference clock having the changed duty cycle while keeping the predetermined frequency is generated (step 14). The low-interference clock can be an asymmetrical clock signal. Since the square wave clock has rising and falling edges, the changing of the duty cycle changes the position of the falling edge of the square wave clock respect to the position of the rising edge of the clock.

Next, a simulation of an exemplary system for removing the third harmonic of the 1 MHz clock on a digital IC is detailed. The system can be applied for suppressing any arbitrary harmonic by choosing the appropriate parameters. Code for performing the simulation in Matlab is shown in the Appendix and pictures of the signal spectrums at 1, 2, 3 KHz are shown in Figures 4-5. Figures 4 and 5 show the spectrum for both cases for $\tau = T/2$, and T/3. In Figures 4-5, the magnitude of the 3 KHz signal is the magnitude of third harmonic signal. The frequency is scaled from 1 MHz to 1 KHz to reduce the sampling rate. FIG. 6 shows a time domain analysis of the 24 KHz clock along with the 1 KHz clock with two different duty cycles.

Viewing Figures 4-6 in combination, changing the duty cycle of a square wave signal will eliminate or suppress different harmonics of that signal on a digital IC. Certain considerations in designing the digital IC and using this method to eliminate or suppress the harmonics for any clock signal on the chip include:

1) Changing the clock duty cycle will change the position of the falling edge of the clock respect to the position of the rising edge of the clock. Thus, if both edges of the clock are used, the change in positions of the edges needs to be considered in the design.

2) Eliminating one harmonic will change the magnitude of the other harmonic. In the example, from the spectrum of the signal, a symmetric clock does not have any second harmonic, but the modified clock with no third harmonic has a significant second harmonic.

Although an illustrative embodiment of the present invention, and various modifications thereof, have been described in detail herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to this precise embodiment and the described modifications, and that various changes and further modifications may be effected therein by one skilled in the art without departing from the scope or spirit of the invention as defined in the appended claims.

APPENDIX

```
%% ******************************* INCLUDE FILE INFO *******************************
%%
%%  Include File Name : harmonic_analysis.m
%%  Original Author   : Masoud Azmoodeh
%%  Date of Origin    : 09/04/2001
%%
%%  COPYRIGHT 2001    - All Rights Reserved     - Airify Communication, Irvine Ca
%%
disp(' ******************************* DESCRIPTION ****************************************');
disp('                                                                                      ');
disp(' This is a matlab routine to analyze the effect of the third harmonics of 1 Mhz clock   ');
disp(' on 3 MHz IF frequency. By changing the tau, the amount of the third harmonic will      ');
disp(' change. For theoretical details, please refer to third_harmonic1.doc and to the Fourier');
disp(' transform of a periodical signal at any communication book.                            ');
disp(' In this simulation, a 1 KHz signal is used instead of 1Mhz to reduce the sampling rate.');
disp('                                                                                      ');
disp('                                                                                      ');
disp(' general shape of the square wave                                                     ');
disp('                                                                                      ');
disp('                                 |                tau                                 ');
disp('                               --|--              ----            ----                ');
disp('                                 | |              |  |            |  |                ');
disp('                                 | |--------------|  |------------|  |                ');
disp('             --------------------|----------------------------------------------------------  ');
disp('                                 |<--------------->|                                  ');
disp('                                 |        T                                           ');
disp('                                 |                                                    ');
disp('                                 |                                                    ');
disp('                                                                                      ');
disp(' Please zoom on the pictures and verify the magnitude of the signal at 1, 2, 3 KHZ range');
disp('                                                                                      ');
disp(' ********************************* Code ********************************************');
%%
%% Lets first generate a symetric square wave with f = 1000 hz clock, sampling frequency=96000
%%
clear;
fs=96000;
f=1000;
T=fs/f;
%%
%%x1=square(2*pi*f/fs*[0:4999]); %%This one needs signal processing toolbox that we do not have now
tau=round( (48/96)*T);               %% Using tau equal to 1/2 * period
x11=[ones(1,tau), zeros(1,T-tau)];
x22 = [ x11 x11 x11  x11 x11 x11  x11 x11 x11  x11 x11 x11  x11 x11 x11  x11 x11 x11  x11 x11 x11 ...
       x11 x11 x11  x11 x11 x11  x11 x11 x11];
x1 = [ x22 x22 x22 ];

fftx1=fft((x1-mean(x1)),fs);
figure(1); hold off; subplot(2,1,1), plot(20*log10( 1e-10+abs(fftx1(1:fs/2)) )); zoom on; grid;
title( ' Spectrum of symmetric square wave, tau = T/2'); xlabel('Hz');
ylabel('20log(|fft()|)');

tau=round( (32/96)*T);               %% Using tau equal to 1/3 * period, will totally get ride of the 3rd
harmonic
x11=[ones(1,tau), zeros(1,T-tau)];
x22 = [ x11 x11 x11  x11 x11 x11  x11 x11 x11  x11 x11 x11  x11 x11 x11  x11 x11 x11  x11 x11 x11 ...
       x11 x11 x11  x11 x11 x11  x11 x11 x11];
x2 = [ x22 x22 x22 ];
fftx2=fft((x2-mean(x2)),fs);
subplot(2,1,2), plot(20*log10( 1e-10+abs(fftx2(1:fs/2)) )); zoom on; grid;
title('Spectrum of asymmetric square wave, with tau=T/3');xlabel('Hz');
ylabel('20log(|fft()|)');
%%
%% Lets try to generate the 1 Mhz clock from 48 Mhz input clock/oscillator
%% First lets generate the 48 Khz clock
tmp0 = [1 1 0 0   1 1 0 0   1 1 0 0   1 1 0 0   1 1 0 0];
tmp1 = [tmp0  tmp0  tmp0  tmp0  tmp0   tmp0  tmp0  tmp0  tmp0  tmp0];
x24  = [tmp1  tmp1  tmp1  tmp1  tmp1   tmp1  tmp1  tmp1  tmp1  tmp1   tmp1  tmp1  tmp1  tmp1  tmp1];

figure(3); subplot(3,1,1); hold off;
subplot(3,1,1); hold off;
plot(x24(1:1.25*96),''); grid on;zoom on;hold on; plot(x24(1:1.25*96),'k.');
title(' The 24 KHz clock/oscillator samples at 96 Ksps along with counter1 data');
t2=0.1
text(1,t2,'10');
%%
tt=5;
text(tt,t2,'9'); text(tt+4,t2,'8'); text(tt+2*4,t2,'7'); text(tt+3*4,t2,'6');
```

```
text(tt+4*4,t2,'5'); text(tt+5*4,t2,'4'); text(tt+6*4,t2,'3');
%%
text(tt+7*4,t2,'2');     text(tt+8*4,t2,'1');      text(tt+9*4,t2,'0');
tt=tt-1;
text(tt+10*4,t2,'23');  text(tt+11*4,t2,'22');    text(tt+12*4,t2,'21');
text(tt+13*4,t2,'20');  text(tt+14*4,t2,'19');    text(tt+15*4,t2,'18');
text(tt+16*4,t2,'17');  text(tt+17*4,t2,'16');    text(tt+18*4,t2,'15');
text(tt+19*4,t2,'14');  text(tt+20*4,t2,'13');    text(tt+21*4,t2,'12');
text(tt+22*4,t2,'11');  text(tt+23*4,t2,'10');
tt=tt+1;   text(tt+24*4,t2,'9');
text(tt+25*4,t2,'8');    text(tt+26*4,t2,'7');     text(tt+27*4,t2,'6');
text(tt+28*4,t2,'5');    %% text(tt+29*4,t2,'4');     text(tt+30*4,t2,'3');
%% text(tt+31*4,t2,'2');    text(tt+32*4,t2,'1');     text(tt+33*4,t2,'0');
%% text(tt+34*4,t2,'23');

%%
subplot(3,1,3); hold off;
tmp1=[ zeros(1,12), x1(1:96*1.25-12)];
plot(tmp1,''); hold; plot(tmp1,'k.'); grid on; zoom on;
title(' The 50/50 percent duty cycle 1 KHz clock generated using 24 KHz oscillator/clock sampled at 96
Kilo samples per second');
%%
subplot(3,1,2); hold off;
tmp2=[ zeros(1,12), x2(1:96*1.25-12)];
plot(tmp2,''); hold on; grid on; zoom on; plot(tmp2,'k.');
title(' The 33/67 percent duty cycle 1 KHz clock generated using 24 KHz oscillator/clock sampled at 96
Kilo samples per second');
%%
```